UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/672,145	09/27/2000	Thomas E. Saulpaugh	5181-67300	6194

7590	07/01/2004	EXAMINER

Robert C Kowert
Conley Rose & Tayon PC
P O Box 398
Austin, TX 78767-0398

EXAMINER
BRANCOLINI, JOHN R

ART UNIT	PAPER NUMBER
2153	7

DATE MAILED: 07/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/672,145	SAULPAUGH ET AL.
	Examiner	Art Unit
	John R Brancolini	2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 April 2004*.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-68* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *55-58* is/are allowed.

6)☒ Claim(s) *1-5,11-28,34-54,59 and 63-68* is/are rejected.

7)☒ Claim(s) *6-10,29-33 and 60-62* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 April 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

This action in response to Amendment filed April 12, 2004.

Drawings

The objection to the drawing is withdrawn by examiner in light of amended Figure 10b.

Information Disclosure Statement

All three information disclosure statements submitted with the original application were reviewed by the examiner, and each PTO 1449 was signed on   The examiner apologizes if these forms were not mailed to the applicant with the initial action. The signed 1449 forms will be resubmitted with this action.

Claim Rejections - 35 USC § 102

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-5, 11-17, 22-28, 34-37, 42-46, 48-52, 54, 59, 63-65, 68 are rejected under 35 U.S.C. 102(e) as being anticipated by Bittenger et al. (US Patent Number 6453362), hereinafter referred to as Bittenger.

In regards to claim 1, Bittenger discloses a method for remotely invoking methods in a distributed computing environment, comprising:

- A client generating a message in a data representation language, wherein the message includes information representing a computer programming language method call, and wherein the message further includes a credential for allowing the client access to a service configured to perform functions on behalf of clients in the distributed computing environment (client issues a command for starting an application, col 7 lines 27-30, a ticket is created as a credential, col 6 lines 63-67).

- The client sending the message to the service (the command is issued to a remote computer, col 7 lines 27-30).

- The service examining the credential included in the message (the ticket is sent along with login info in the first message for the server to examine, col 7 lines 10-21).

- If said examining determines the credential is authentic, the service performing a function on behalf of the client in accordance with the information representing the computer programming language method call included in the message (after checking the ticket, application is started, col 7 lines 32-36).

- If said examining determines the credential is not authentic, the service not performing the function on behalf of the client (if validation does not occur, no operation is performed).

In regards to claim 2, Bittenger discloses the client comprises a client method gate configured to provide an interface to the service by generating data representation language messages including information representing method calls, and wherein said generating a message is performed by the client method gate (after receiving validation, the ticket acts as a gate to generate messages, col 7 lines 50-57).

In regards to claim 3, Bittenger discloses the sending the message is performed by the client method gate (the ticket is used in the creation of a server stub which is used to send messages and requests, col 7 lines 55-57).

In regards to claim 4, Bittenger discloses the client further comprises a client process, the method further comprising:

- The client process generating the computer programming language method call (ticket generates a method call, col 7 lines 50-57).

- The client method gate receiving the method call generated by the client process (The server stub responds to the method call, col 7 lines 50-57).

- Wherein said generating a message is performed in response to said receiving the method call (The server stub creates requests for the application, col 7 lines 55-57).

In regards to claim 5, Bittenger discloses the client further comprises a client message endpoint, wherein said sending the message to the service comprises:

- The client method gate sending the message to the client message endpoint, wherein the client message endpoint is configured to send messages in the data representation language to the service (The client ticket acts as a gate sending the message to the server stub, col 7 lines 50-57).

- The client message endpoint attaching the credential to the message (tStamp is an identifier used on all messages, col 7 lines 1-5).

- The client message endpoint sending the message to the service (the server stub sends the request to the server, col 7 lines 55-57).

In regards to claim 11, Bittenger discloses the service comprises a service message endpoint configured to receive messages in the data representation language from the client, wherein said performing a function comprises the service message endpoint receiving the message from the client (the server stub, originally sent to the client, is re-generated by the client to act as an endpoint, col 7 lines 32-57).

In regards to claim 12, Bittenger discloses the service comprises one or more computer programming language methods executable within the service, wherein said performing a function comprises executing a computer programming language method of the service in accordance with the information representing the computer

programming language method call included in the message (the server receives the message call which is a request of functions to be performed, col 8 lines 29-45).

In regards to claim 13, Bittenger discloses the service comprises one or more computer programming language methods executable within the service, wherein the information representing the computer programming language method call includes an identifier of the method call, and wherein said performing a function comprises:

- Regenerating the method call in accordance with the identifier of the method call included in the information representing the method call (the ticket uses a server stub and tStamp as an identifier to represent the method call, col 7 lines 1-9)

- Executing a computer programming language method of the service in accordance with the regenerated method call (the server stub passes the message call to the server for execution, col 7 lines 27-49).

In regards to claim 14, Bittenger discloses the information representing the computer programming language method call further includes one or more parameter values of the method call, and wherein said executing a computer programming language method in accordance with the regenerated method call comprises providing the one or more parameter values from the information representing the method call as parameter values of the method call (The server stub acts a set of parameters followed when requesting data, col 7 lines 41-57).

In regards to claim 15, Bittenger discloses the service further comprises a service method gate configured to provide an interface to the one or more computer programming language methods of the service by receiving data representation language messages and invoking computer programming language methods specified by the messages, and wherein said regenerating the method call is performed by the service method gate (server stub is used as a gate to provide an interface to one or more computer programming language methods, col 7 lines 41-57).

In regards to claim 16, Bittenger discloses performing a function generates results data, the method further comprising the service providing the generated results data to the client (the requests are used to create a custom process that will provide generated results to the client, col 8 lines 32-45).

In regards to claim 17, Bittenger discloses performing a function generates results data, and wherein the service comprises a service message endpoint configured to send messages in the data representation language to the client for the service, the method further comprising:

- The service message endpoint sending a results message to the client, wherein the results message includes the generated results data (the requests are used to create a custom process that will provide generated results to the client, col 8 lines 32-45).

In regards to claim 22, Bittenger discloses the computer programming language is the Java programming language, and wherein the information representing the method call in the message represents a Java method call to a Java method implemented on the service, and wherein the service performing a function comprises invoking the Java method on the service in accordance with the information representing the Java method call included in the message (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 23, Bittenger discloses the client is executing within a virtual machine, wherein the virtual machine is executing within a client device in the distributed computing environment (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 24, Bittenger discloses the virtual machine is a Java Virtual Machine (JVM) (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 25, Bittenger discloses a distributed computing system comprising:

- A service device comprising one or more functions executable on the service device on behalf of client devices in the distributed computing system (Fig 1 shows an outline of the system, including multiple computers and a server side application).

- A client device configured to:

  - Generate a message in a data representation language, wherein the message includes information representing a computer programming language method call, and wherein the message further includes a credential for allowing the client device access to the service device (client issues a command for starting an application, col 7 lines 27-30, a ticket is created as a credential, col 6 lines 63-67).

  - Send the message to the service device (the command is issued to a remote computer, col 7 lines 27-30).

- Wherein the service device is configured to:

  - Examine the credential included in the message (the ticket is sent along with login info in the first message for the server to examine, col 7 lines 10-21).

  - If said examining verifies the credential, perform a function on behalf of the client in accordance with the information representing the computer programming language method call included in the message (after checking the ticket, application is started, col 7 lines 32-36).

  - If said examining does not verify the credential, not perform the function on behalf of the client (if validation does not occur, no operation is performed).

In regards to claim 26, Bittenger discloses the client device comprises a client method gate configured to provide ail interface to the service by generating data representation language messages including information representing method calls, and wherein said generating a message is performed by the client method gate (after receiving validation, the ticket acts as a gate to generate messages, col 7 lines 50-57).

In regards to claim 27, Bittenger the client device further comprises a client process,

- Wherein the client process is configured to generate the computer programming language method call (ticket generates a method call, col 7 lines 50-57).

- Wherein the client method gate is further configured to receive the method call generated by the client process (The server stub responds to the method call, col 7 lines 50-57).

- Wherein said generating a message is performed by the client method gate in response to said receiving the method call (The server stub creates requests for the application, col 7 lines 55-57).

In regards to claim 28, Bittenger the client device further comprises a client message endpoint,

- Wherein the client method gate is further configured to send the message to the client message endpoint (The client ticket acts as a gate sending the message to the server stub, col 7 lines 50-57).

- Wherein the client message endpoint is configured to:

  - Attach the credential to the message (tStamp is an identifier used on all

    messages, col 7 lines 1-5).

  - Send the message to the service device (the server stub sends the

    request to the server, col 7 lines 55-57).

In regards to claim 33, Bittenger discloses the client device is further configured

to:

- Generate a client message endpoint in accordance with the service

  advertisement, wherein the client message endpoint is configured to send the

  data representation language messages to the address (the server stub,

  originally sent to the client, is re-generated by the client to act as an endpoint, col

  7 lines 32-57).

- Wherein said sending the message to the service device is performed by the

  client message endpoint (the server stub generates the request, col 7 lines 55-

  57).

In regards to claim 35, Bittenger discloses the information representing the

computer programming language method call further includes one or more parameter

values of the method call, and wherein, in said executing a computer programming

language method in accordance with the regenerated method call, the service device is

further configured to:

- Provide the one or more parameter values from the information representing the

  method call as parameter values of the method call (The server stub acts a set of

  parameters followed when requesting data, col 7 lines 41-57).

In regards to claim 36, Bittenger discloses the service device further comprises a

service method gate configured to provide an interface to the one or more computer

programming language methods of the service by receiving data representation

language messages and invoking methods specified by the messages, and wherein

said regenerating the method call is performed by the service method gate (server stub

is used as a gate to provide an interface to one or more computer programming

language methods, col 7 lines 41-57).

In regards to claim 37, Bittenger discloses performing a function generates

results data, wherein the service device comprises a service message endpoint

configured to send a results message in the data representation language to the client

device, wherein the results message includes the generated results data (the requests

are used to create a custom process that will provide generated results to the client, col

8 lines 32-45).

In regards to claim 42, Bittenger discloses the computer programming language

is the Java programming language, and wherein the information representing the

method call in the message represents a Java method call to a Java method

implemented on the service, and wherein, in said performing a function, the service

device is further configured to invoke the Java method on the service device in

accordance with the information representing the Java method call included in the

message (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64

– col 5 line 7).

In regards to claim 43, Bittenger discloses:

- A virtual machine executable within the client device.

- A client process executable within the virtual machine, wherein said generating a
  message and said sending the message are performed by the client process (the
  system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5
  line 7).

In regards to claim 44, Bittenger discloses the virtual machine is a Java Virtual

Machine (JVM) (the system is written in Java and utilizes a Java Virtual Machine, col 4

line 64 – col 5 line 7).

In regards to claim 45, Bittenger discloses a device comprising:

- A client component (Fig 1 item 11 shows the client).

- A method gate (after receiving validation, the ticket acts as a gate to generate
  messages, col 7 lines 50-57).

- Wherein the client component is configured to generate a computer programming language method call (ticket generates a method call, col 7 lines 50-57).

- Wherein the method gate is configured to:

  o Access the computer programming language method call generated by the client component (client issues a command for starting an application, col 7 lines 27-30, a ticket is created as a credential and as a method generator, col 6 lines 63-67).

  o Generate a message in a data representation language, wherein the message includes information representing a computer programming language method call, and wherein the message further includes a credential for allowing the client device access to a service in a distributed computing environment (after receiving validation, the ticket acts as a gate to generate messages, and creates a server stub to send the messages, col 7 lines 50-57).

  o Send the message to the service (the server stub sends the request to the server, col 7 lines 55-57).

- Wherein the service is operable to verify the message as authentic by examining the credential included in the message, and to perform a function on behalf of the client component in accordance with the information representing the computer programming language method call included in the message if the message is verified as authentic (the ticket is sent along with login info in the first message

for the server to examine, col 7 lines 10-21, after checking the ticket, application is started, col 7 lines 32-36)..

In regards to claim 46, Bittenger discloses the method gate comprises a data representation language message schema comprising descriptions of data representation language messages the device is authorized to send to the service, wherein said generating a message is performed in accordance with a description of the message comprised in the message schema (the server stub is sent to the client as a parameter to define messages and requests the client can make, col 7 lines 32-30).

In regards to claim 48, Bittenger discloses the computer programming language is the Java programming language, and wherein the information representing a method call in the message represents a Java method call to a Java method implemented on the service (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 49, Bittenger discloses a device comprising:

- A client component configured to generate a message in a data representation language, wherein the message includes information representing a computer programming language method call (client issues a command for starting an application, col 7 lines 27-30, a ticket is created as a credential and as a method generator, col 6 lines 63-67).

- A message endpoint configured to:

  o Attach a credential to the message for allowing the client component

     access to a service in a distributed computing environment (a ticket is

     created as a credential, col 6 lines 63-67).

  o Send the message to a service in a distributed computing environment

     (the server stub sends the request to the server, col 7 lines 55-57).

- Wherein the service is operable to verify the message as authentic by examining

  the credential included in the message, and to perform a function on behalf of the

  client component in accordance with the information representing the computer

  programming language method call included in the message if the message is

  authentic (the ticket is sent along with login info in the first message for the

  server to examine, col 7 lines 10-21, after checking the ticket, application is

  started, col 7 lines 32-36)..

In regards to claim 50, Bittenger discloses the client component is further

configured to generate the computer programming language method call, and wherein

said generating a message is performed in response to said generating the computer

programming language method call (ticket generates a method call, col 7 lines 50-57).

In regards to claim 51, Bittenger discloses the device further comprises a virtual

machine executable within the device, wherein the client component and the message

endpoint are executable within the virtual machine (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 52, Bittenger discloses the virtual machine is a Java Virtual Machine (JVM) (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

In regards to claim 54, Bittenger discloses the computer programming language is the Java programming language, and wherein the information representing a method call in the message represents a Java method call to a Java method implemented on the service.

In regards to claim 59, Bittenger discloses a carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement:

- A client generating a message in a data representation language, wherein the message includes information representing a computer programming language method call, and wherein the message further includes a credential for allowing the client access to a service configured to perform functions on behalf of clients in the distributed computing environment (client issues a command for starting an application, col 7 lines 27-30, a ticket is created as a credential, col 6 lines 63-67).

- The client sending the message to the service (the server stub sends the request to the server, col 7 lines 55-57).

- The service examining the credential included in the message (the ticket is sent along with login info in the first message for the server to examine, col 7 lines 10-21).

- If said examining determines the credential is authentic, the service performing a function on behalf of the client in accordance with the information representing the computer programming language method call included in the message (after checking the ticket, application is started, col 7 lines 32-36).

- If said examining determines the credential is not authentic, the service not performing the function on behalf of the client (if validation does not occur, no operation is performed).

In regards to claim 63, Bittenger discloses the service comprises one or more computer programming language methods executable within the service, wherein the information representing the computer programming language method call includes an identifier of the method call, and wherein, in said performing a function, the program instructions are further computer-executable to implement:

- Regenerating the method call in accordance with the identifier of the method call included in the information representing the method call (the ticket uses a server stub and tStamp as an identifier to represent the method call, col 7 lines 1-9).

- Executing a computer programming language method of the service in
  accordance with the regenerated method call (the server stub passes the
  message call to the server for execution, col 7 lines 27-49).

In regards to claim 64, Bittenger discloses the information representing the
computer programming language method call further includes one or more parameter
values of the method call, and wherein, in said executing a computer programming
language method in accordance with the regenerated method call, the program
instructions are further computer-executable to implement providing the one or more
parameter values from the information representing the method call as parameter
values of the method call (The server stub acts a set of parameters followed when
requesting data, col 7 lines 41-57).

In regards to claim 65, Bittenger discloses the service further comprises a service
method gate configured to provide an interface to the one or more computer
programming language methods of the service by receiving data representation
language messages and invoking computer programming language methods specified
by the messages, and wherein said regenerating the method call is performed by the
service method gate (server stub is used as a gate to provide an interface to one or
more computer programming language methods, col 7 lines 41-57).

In regards to claim 68, Bittenger discloses the computer programming language is the Java programming language, and wherein the information representing the method call in the message represents a Java method call to a Java method implemented on the service, and wherein, in said performing a function, the program instructions are further computer-executable to implement invoking the Java method on the service in accordance with the information representing the Java method call included in the message (the system is written in Java and utilizes a Java Virtual Machine, col 4 line 64 – col 5 line 7).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 18-20, 38-40, 47, 53, 66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bittenger in view of Leach et al. (US Patent Number 6108715).

In regards to claims 20, 40, and 66, Bittenger fails to disclose that a URI, or defined addressing space, is provided for accessing stored data. Leach however, discloses that an address is provided for accessing the data stack, and the types of files are known by the client, which are the key features of a URI. The addressing space is provided to allow the user to immediately access the data at the provided address.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify Bittenger to include providing a defined address space, or a URI to the client as taught by Leach to allow the user to immediately access the data at the provided address.

In regards to claims 18-19, 38-39, Bittenger discloses all elements of the current claims (see above discussions), however, Bittenger lacks providing a storage space for results as well as informing the user where to access the stored results. Leach discloses the creation of a data stack used to store the results of the operations locally on the server, then allows the clients to map back to the stored results stack by providing an address which allows direct transfer between the client and the server greatly reducing the processing overhead (col 3 lines 18-30).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify Bittenger to include providing a storage space for results as well as informing the user where to access the stored results as taught by Leach to allows direct transfer between the client and the server greatly reducing the processing overhead.

In regards to claims 47, 53, Bittenger fails to disclose providing the client access to the stored results data. Leach discloses the creation of a data stack used to store the results of the operations locally on the server, then allows the clients to map back to the

stored results stack by providing an address which allows direct transfer between the

client and the server greatly reducing the processing overhead (col 3 lines 18-30).

. It would have been obvious to one of ordinary skill in the art at the time of

invention to modify Bittenger to include providing the client access to the stored results

data as taught by Leach to allows direct transfer between the client and the server

greatly reducing the processing overhead.

Claims 21, 41, and 67 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Bittenger in view of the Instaweb Online Computing Dictionary

(Instaweb, http://www.instantweb.com/foldoc/foldoc.cgi?query=XML)

Bittenger fails to disclose the data representation language is XML. Instaweb,

however, shows that XML can be used to create custom tags for data objects that offer

greater flexibility in organizing and presenting information.

It would have been obvious to one of ordinary skill in the art at the time of

invention to modify Bittenger to use XML as the data representation language as taught

by Instaweb to allow for the creation of custom tags for data objects that offer greater

flexibility in organizing and presenting information.

Allowable Subject Matter

Claims 55-58 allowed.

Claims 6-10, 29-33, 60-62 are objected to as being dependent upon a rejected

base claim, but would be allowable if rewritten in independent form including all of the

limitations of the base claim and any intervening claims.

Response to Arguments

Applicants arguments towards 102(e) rejections of claims 1, 25, 45, 49, and 59:

1. Bittenger fails to teach the use of a data representation language.

2. Bittenger fails to teach a client attaching a credential to the message.

In regards to argument 1, the examiner respectfully disagrees. From solely

reading the claim language, a data representation language seems to be nothing more

than a method of encoding data for transmission between a client and a server where

both the transmitter and receiver have means of reading the data encoded in the

message. Bittenger clearly has this feature.

In regards to argument 2, the examiner respectfully disagrees with applicant.

Reading solely from the claim language, a credential is attached top the first message

the client transmits to the server, along with the varied parameters for the request.

Bittenger, col 7 lines 58 – col 8 line 31. In this passage, Bittenger shows the login and

verification process, in which the user sends application parameters along with all

necessary credential information. Once the credential information is evaluated, the

parameters are passed to the application server for configuring for the client requests.

This is seen as the client generating a message including information representing a computer programming language call (the passed request parameters) as well as a credential allowing the client to access a service configured to perform functions on behalf of the client.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to John R Brancolini whose telephone number is (703) 305-7107. The examiner can normally be reached on M-Th 7am-5:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess can be reached on (703) 305-4792. The fax phone
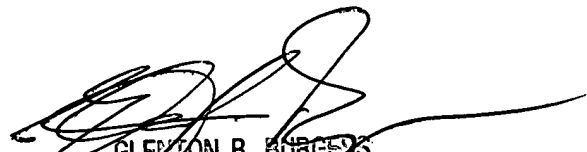
number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JRB

GLENNON B. BURGESS
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100